# The Aspect of Data Translation in Service Similarity

# The Aspect of Data Translation in Service Similarity

Dionysis Athanasopoulos
*School of Engineering & Computer Science*
*Victoria University of Wellington, New Zealand*
*dionysis.athanasopoulos@ecs.vuw.ac.nz*

***Abstract*—To avoid the lock-in problem in service-oriented software, decoupling mechanisms have been proposed that identify service and schema high-level mappings for developing service-translation scripts. However, based on the fundamental data-translation process, the most important step is to produce low-level mappings that satisfy schema constraints. The problem is that the production of such mappings may not be feasible if service schemas are highly heterogeneous. Thus, we propose a proactive approach that firstly identifies similar services, which satisfy schema constraints, and then, provides them to decoupling mechanisms. In particular, given schema constraints, our approach follows a composite workflow for estimating service-translation cost, which reflects an upper-bound of the ensured schema constraints. We evaluate the effectiveness of our approach against a state-of-the-art service similarity approach and the results show that high service similarity does not necessarily imply low service-translation cost, the bidirectional nature of service similarity can be misleading, ensured schema constraints improve service similarity, our estimated translation cost is very close to the corresponding actual cost, and service retrieval and clustering mechanisms are effective when they use our approach.**

***Keywords*-service-oriented software; service similarity; data translation; schema constraints; decoupling; composite metric;**

## I. Introduction

Famous vendors (e.g. `Google`, `Amazon`) make available their resources as services via using the Web-service technology[1]. However, software built on top of these services, using the Service-oriented Architecture (*SoA*) style, should not be developed with respect to APIs (Application Programming Interfaces) of specific service providers, since it may be locked in outdated software functionality and/or data schemas. Moreover, the high service heterogeneity makes the migration of *SoA* software too difficult. Since Web services have become one of the standard technologies on the Web (esp. on the cloud), the lock-in problem is topical. To avoid from *interface perspective* the lock-in problem, the general idea is to decouple *SoA* software from specific APIs, avoiding to create separate clients for every provider. Decoupling (e.g. adapter-based) mechanisms map service interfaces and translate their invocations.

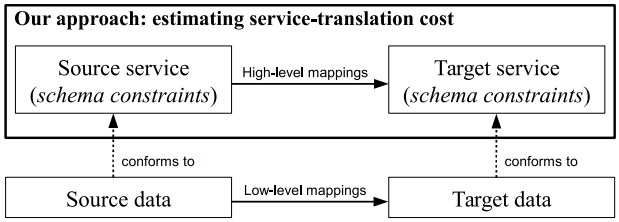
**Motivation**. Existing interface-decoupling mechanisms propose categories of mismatches for identifying service/ schema mappings and developing service-translation scripts (e.g. recent survey in [1]). Service translation *at schema level* is a variation of the traditional *data-translation* problem [2], in which the identification of bidirectional 1–1 (high-level) mappings is only the first step. The next and most important step is to produce directed low-level mappings that satisfy *schema* (*integrity* and *structural*) *constraints* (Fig. 1 (a)) [3]. They are directed, since the translation process is not necessarily communitative (i.e. the reverse translation may not be feasible or performed at the same steps) and low-level, since they are used for translating *actual* values.

The problem is that the production of low-level mappings that satisfy schema constraints may not be feasible if service schemas are highly heterogeneous. The impact of constraint violations is the possible unsuccessful outcome of service translation. Thus, *what is still missing in the literature is a proactive solution that identifies similar services via taking into account the aspect of data translation*.

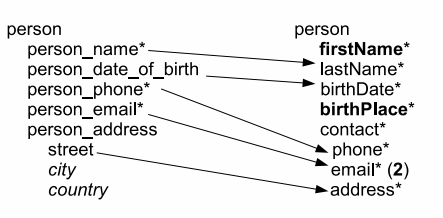
As a simple, motivating, and illustrative example, Fig. 1 (b) depicts the high-level mappings between a part of the message schemas of two (person registration) services. We observe that the elements `firstName` and `birthPlace` of the target schema, whose values are necessary (star notation), are unmapped and one occurrence of the target element `email` is also unmapped, since its source element occurs only once. Overall, missing values in elements related to constraints (bold font) make unsuccessful the (automation of the) translation process.

**Contribution**. We propose an approach that firstly identifies similar services of low translation cost and following, provides them to decoupling mechanisms. In particular, given schema constraints and a set of high-level mappings, our approach estimates service-translation cost, which reflects an upper-bound of the satisfied schema constraints (Fig. 1 (a)). Our approach follows a *composite* workflow (instead of hybrid, typically used in service similarity), which comprises the following steps: (i) service pre-processing for determining schema constraints, (ii) high-level mapping identification, and (iii) service-translation cost estimation. Composite workflows offer clear interpretation of their results [4] and are more effective than hybrid ones [5]. Our contribution mainly focuses on the third workflow step, since the second step has already been investigated by the literature. We also contribute in the first step, since schema constraints are not specified by service-interface documents.

[1] https://www.w3.org/TR/ws-arch




(a) Our approach in the service-translation process.




(b) High-level mappings between the schemas of two service messages.

Figure 1. The service-translation process and schema constraints violated by high-level mappings.

To evaluate the effectiveness of our approach, we implement two versions of service retrieval and clustering mechanisms, one for our approach and one for a state-of-the-art service-similarity approach. The results on the Web services of a benchmark show that (i) high service similarity does not necessarily imply low service-translation cost; (ii) the bidirectional nature of service similarity can be misleading; (iii) ensured schema constraints improve service similarity; (iv) our estimated service-translation cost is very close to the corresponding actual cost; (v) service retrieval and clustering mechanisms are effective using our approach.

Overall, our contribution is summarized as follows:

- we categorize and compare state-of-the-art approaches
- we specify the notion of schema constraints
- we propose a metric for estimating translation cost
- we specify the composite workflow
- we evaluate our approach on benchmark services.

The rest of the paper is structured as follows. Section II presents the state-of-the-art. Section III defines the basic notions. Section IV defines the proposed metric. Section V describes the composite workflow. Section VI presents the evaluation of our approach. Finally, Section VII summarizes our approach and discusses its future research directions.

## II. Related Work

Our work belongs to the research domains of service similarity and *SoA* decoupling. We do not describe data-translation approaches, since they solely produce translation scripts, without estimating translation cost.

### A. Service Similarity

From functional (esp. interface) perspective, many similarity approaches (included recent ones) compare *keywords/terms* in service interfaces (e.g. [6], [7], [8], [9]). Other approaches compare operation *signatures* (e.g. [10], [11], [12]). In addition to operation signatures, [13] compares the documentation of service operations. Other approaches match message schemas (e.g. [14], [15]). Finally, [16] and [17] identify similar services via determining relaxed versions of the object-oriented notion of the *behavioral subtyping* [18] and identifying high-level mappings ([16] in the case of service versions, while [17] in the case of *autonomous* services, i.e. developed by different providers).

**Comparison**. We compare the approaches in terms of the following criteria: service relation, similarity metric, adopted technique, and translation aspect. Based on their comparison (Table I), *our approach is the (i) only similarity approach of composite workflow that (ii) takes into account schema constraints for estimating translation cost*.

Table I
Summary and comparison of service-similarity approaches.

| | Service Relation | Similarity Metric | Technique | Translation Aspect |
|---|---|---|---|---|
| [6]-[9] | similarity (autonomous) | hybrid | keywords | X |
| [11] [10] [12] | similarity (autonomous) | hybrid | signatures | X |
| [13] | similarity (autonomous) | hybrid | keywords & signatures | X |
| [14], [15] | similarity (autonomous) | hybrid | schema matching | X |
| [16] | subtyping (versioning) | hybrid | schema matching | X |
| [17] | subtyping (autonomous) | hybrid | schema matching | X |
| *Ours* | *subtyping (autonomous)* | *composite* | *schema constraints* | ✓ |

### B. Decoupling in SoA Software

*SoA* decoupling mechanisms are based on adapter/proxy patterns [19] or service abstractions. Adapter provides different interface to that of the adapted services, while proxy the same interface. Abstraction generizes concrete services. The mechanisms range from *static*, realized at compile time, to *dynamic*, where a single client is used at runtime.

One of the earliest proxy-based approaches [20] proposes that a provider releases not only service versions, but also their proxies (static decoupling). [21] constructs a graph of dependencies between service versions, used by proxy for dispatching service invocations, manipulating specific categories of service changes (semi-dynamic decoupling).

In the adapter-based approach of [22], the adapter interface is derived from the same popular interface, while in [23], this assumption is relaxed. [24] and [25] define adapters, in which service mismatches are resolved in combination with human intervention (semi-dynamic decoupling). In [26], adapter interface has subtyping relation with

concrete services. [27] and [28] use *has-a* abstractions, while [17], [29], and [30] *is-a* abstractions.

**Comparison**. We compare the approaches in terms of the following criteria: service relation, decoupling type, decoupling technique, and translation aspect. Based on their comparison (Table II), *our approach is the only dynamic decoupling approach that examines service subtyping, taking into account service-translation cost*.

Table II
SUMMARY AND COMPARISON OF APPROACHES FOR SOA DECOUPLING.

| | Service Relation | Decoupling Type | Decoupling Technique | Translation Aspect |
|---|---|---|---|---|
| [20] [21] [22] | similarity (versioning) | static | proxy | X |
| | | semi-dynamic | | |
| [23] [24] [25] [27] [28] | similarity (autonomous) | dynamic | adapter | |
| | | semi-dynamic | | |
| [29], [30] | | dynamic | has-a abstractions | |
| | | | is-a abstractions | |
| [26] | subtyping (autonomous) | | adapter | |
| [17] | | | is-a abstractions | |
| *Ours* | *subtyping (autonomous)* | *dynamic* | *adapter* | ✓ |

## III. BASIC NOTIONS

We define the conceptual models of service interface, schema constraint, and high-level mapping (Sections III-A, III-B, and III-C). We also introduce the notion of service-translation cost, which is formally defined in Section IV.

### A. Service Interface

According to our model, derived by WSDL-based services[2], a service interface, `I` (Table III (Eq. 1)), is characterized by its `name` and set of operations, `OPS` (Table III (Eq. 2)). An operation, `OP` (Table III (Eq. 3)), accepts an input, `in`, and produces an output message, `out`. A message, `MSG` (Table III (Eq. 4)), comprises its `name` and set of elements of built-in data-types, `E` (Table III (Eq. 5)), which correspond to leaf elements of the tree structure of an XML[3] message-schema. We consider only leaf elements, since they are the places where actual values are inserted during translation.

We define a leaf element as a tuple (Table III (Eq. 5)) that consists of its `name`, built-in `type` (subtype of `anyType`[3]), and interval, `OC` (Table III (Eq. 6)), of its `min` and `max` occurrence numbers. We assume that the min (resp. max) occurrences number of a leaf element equals to the product of the min (resp. max) occurrence numbers of all the elements that lie on the path from the root to the leaf element, since we need an upper-bound of the mapped element-instances. We further assume that a leaf element

[2]https://www.w3.org/TR/wsdl

[3]https://www.w3.org/XML

Table III
THE DEFINITION OF THE PROPOSED SERVICE-INTERFACE MODEL

$$I := (name : \texttt{String},\ OPS) \quad (1)$$

$$OPS := \{op_i : OP\} \quad (2)$$

$$OP := (name : \texttt{String},\ in : MSG,\ out : MSG) \quad (3)$$

$$MSG := (name : \texttt{String},\ keys : E,\ com : E,\ rest : E) \quad (4)$$

$$E := \Big\{(name : \texttt{String},\ type : \texttt{anyType},\ oc : OC)\Big\} \quad (5)$$

$$OC := [min \in \mathbb{N}, max \in \mathbb{N}] \quad (6)$$

inherits the *order indicator* of its parent, which can be one of the following: `all` (i.e. its children appear in any order but at most once), `sequence` (i.e. only one of its children occurs), or `choice` (i.e. its children must appear in the specified order). Finally, the leaf elements of a message are organized into the categories, `com`, `keys`, and `rest` (Table III (Eq. 4)), by taking into account schema (structural and integrity) constraints, which are explained below.

**Structural constraints**. They are related to parent/child relationships, order constraints, and element appearance [3]. Since we use only leaf elements, we do not examine parent/child relationships. The order constraints of an element is related to its inherited order indicators. Regarding the appearance of an element, it can be optional or its occurrences number is specified using the `minOccurs` and `maxOccurs` attributes. Based on these constraints, we define the notion of compulsory element as follows.

*Definition 1:* An element is `com`pulsory (member of the `com` category) if (i) it is not optional, (ii) it has built-in data-type, (iii) its order indicator is `sequence`, and (iv) its occurrences number is at least one. □

Returning to the example of Fig. 1 (b), the elements, `phone`, `email`, and `address`, are compulsory.

**Integrity constraints**. They are related to primary/foreign key relationships. Primary key in XML refers to an element (or a combination of elements), whose value is unique, non-nullable, and identifies the values of the other elements, which are located at the same layer of the schema tree. While there is a dedicated element in XML for defining keys, it is rarely used in practice[4]. Thus, we assume an element plays the role of primary key when the following holds.

*Definition 2:* An element is primary key (member of the `key` category) if (i) it is compulsory, (ii) its min and max occurrence numbers equal to one, (iii) it has at least one non-leaf sibling element, and (iv) its value is unique. □

Returning to the example of Fig. 1 (b), the combination of the elements, `firstName`, `lastName`, `birthDate`, and `birthPlace`, is the primary key.

The foreign-key notion refers to an element, whose value points to a key element, and is defined as follows.

[4]In the used benchmark, no schema contains such an element.

Table IV
THE DEFINITION OF THE PROPOSED MODEL OF HIGH-LEVEL MAPPINGS.

$$M_I := \big(i_s : I,\ i_t : I,\ m_{OPS} : M_{OPS}\big) \quad (1)$$

$$M_{OPS} := \{m_{OP} : M_{OP}\} \quad (2)$$

$$M_{OP} := \big(op_s : OP,\ op_t : OP,\ m_{IN} : M_{MSG},\ m_{OUT} : M_{MSG}\big) \quad (3)$$

$$M_{MSG} := \big(msg_s : MSG,\ msg_t : MSG,\ m_{KEYS} : M_E,\ m_{COM} : M_E,\ m_{REST} : M_E\big) \quad (4)$$

$$M_E := \Big\{\big(e_s : E,\ e_t : E,\ loss : \texttt{double} \in [0,1]\big)\Big\} \quad (5)$$

*Definition 3:* An element is foreign key (member of the `key` category) if (i) its min and max occurrence numbers equal to one, (ii) it has built-in data-type, (iii) its order indicator is `sequence`, (iv) its name equals to a key name. □

For simplicity reasons, we hereafter use the term key to refer to either a primary or foreign key.

### B. High-level Mappings

We propose the conceptual model of mappings (Table IV), which do not necessarily satisfy schema constraints. We assume the output of the used high-level mapping tool conforms to this model. The model is hierarchically structured, based on our service-interface model, as explained below.

**Interface mapping.** An interface mapping, $M_I$ (Table IV (1)), consists of the source and target service interfaces and a set of operation mappings, $M_{OPS}$ (Table IV (2)).

**Operation mapping.** An operation mapping, $M_{OP}$ (Table IV (3)), consists of the source and target operations and the mappings between their input and output messages.

**Message mapping.** A message mapping, $M_{MSG}$ (Table IV (4)), consists of the source and target messages and the $1-1$ mappings between their leaf elements, divided into three sets: mappings between their key elements, $m_{KEYS}$, compulsory elements, $m_{COM}$, and rest elements, $m_{REST}$.

**Element mappings.** Each element mapping of the set, $M_E$ (Table IV (5)), consists of the source and target elements, along with the amount of information loss when the source value is translated to a target value. The notion of information loss is detailed in Section III-C.

### C. Service-Translation Cost

As discussed, service translation works on low-level mappings. A low-level mapping relates the elements (not only the leaves) of a path of the source schema to the elements of a path of the target schema, using the integrity and structural schema-constraints. For each low-level mapping, a service-translation script performs two steps: the retrieve and insert steps. In the first step, the script traverses a source path and retrieves actual values. In the second step, guided by the mapped path elements, it inserts translated values.

Based on the general description of a script, we estimate the service-translation cost in terms of the following cases: (i) leaf elements of a target path with no retrieved values, (ii) leaf elements of a target path with information loss, and (iii) unmapped leaf elements of a source path.

If the elements of the first or second category are keys or compulsory, then the service-translation cost is high, since it is risky to generate automatically missing values or infeasible to ask by the end-user hundreds of values for large schemas. Otherwise, the service-translation cost is low, since their translation may lead to information loss, which is not though critical. The information loss for built-in data-types (e.g. converting a `double` to an `int`) has been quantified in [13]. Finally, in the third category, the service-translation cost is also low, since some of the source values shall be discarded. For instance, in the example of Fig. 1 (b), the values of the elements, `city` and `country` (italics font), are discarded. Overall, since the actual numbers of missing or translated values cannot be known a priori, we estimate the cost based on upper-bounds, as explained in Section IV.

## IV. ESTIMATING THE COST OF SERVICE TRANSLATION

To estimate the cost for translating invocations from a source to a target service-interface, we propose the $\mathcal{T}$ranslation $\mathcal{C}$ost ($\mathcal{TC}$) metric (Table V (Eq. 1)), which accepts a set of high-level mappings, $m_I$, and the value of a threshold, $\theta$ (corresponds to the min accepted cost value).

### A. Metric of Service-Translation Cost

The metric is hierarchically defined, including a different metric for each layer. The metric values belong to the interval [0, 1] (the value 1 corresponds to the highest cost). To integrate the metric values, we use proper aggregation functions.We also propose a new family of functions in order to cover a specific case of aggregation.

**Metric of interface-translation cost**. We propose the metric, $\mathcal{TC}_I$ (Table V (Eq. 2)), which aggregates the translation costs of their mapped operations. We assume that it is high if the translation costs of their mapped operations are high and the number of their mapped operations is also high. Thus, it is defined as the sum of the translation costs of their mapped operations, divided by the maximum number of the operations (not only the mapped).

**Metric of operation-translation cost**. We propose the metric, $\mathcal{TC}_{OP}$ (Table V (Eq. 3)), which aggregates the translation costs of their input and output messages via using the product aggregation-function, assuming that the operation translation-cost is high if both translation costs of their input and output messages are high.

**Metric of message-translation cost**. We propose the metric, $\mathcal{TC}_{MSG}$ (Table V (Eq. 4)), which calculates the complement of the aggregated percentages of the satisfied integrity ($\mathcal{INT}$) and structural ($\mathcal{STR}$) constraints and of

Table V
THE DEFINITION OF THE PROPOSED METRIC FOR ESTIMATING SERVICE-TRANSLATION COST

$$\mathcal{TC}_I(m_I : M_I, \theta : double) := \begin{cases} \mathcal{TC}_I(m_I) & \text{if } \mathcal{TC}_I(m_I) \geq \theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathcal{TC}_I(m_I : M_I) := \frac{\sum_{k=1}^{|m_I.m_{OPS}|} \mathcal{TC}_{OP}(m_I.m_{OPS}.m_{OP_k})}{MAX\Big(|m_I.i_s.ops|, |m_I.i_t.ops|\Big)} \quad (2)$$

$$\mathcal{TC}_{OP}(m_{OP} : M_{OP}) := \mathcal{TC}_{MSG}(m_{OP}.m_{IN}) * \mathcal{TC}_{MSG}(m_{OP}.m_{OUT}) \quad (3)$$

$$\mathcal{TC}_{MSG}(m_{MSG} : M_{MSG}) := 1 - \mathcal{F}_{PR}\Big(\mathcal{INT}(m_{MSG}) * \mathcal{STR}(m_{MSG}),\ \ \mathcal{IT}(m_{MSG})\Big) \quad (4)$$

$$\mathcal{INT}(m_{MSG} : M_{MSG}) := \frac{|m_{MSG}.m_{KEYS}|}{|m_{MSG}.msg_t.keys|} \quad (5)$$

$$\mathcal{STR}(m_{MSG} : M_{MSG}) := \frac{|m_{MSG}.m_{COM}|}{|m_{MSG}.msg_t.com|} * \frac{|m_{MSG}.m_{REST}|}{|m_{MSG}.msg_t.rest|} \quad (6)$$

$$\mathcal{IT}(m_{MSG} : M_{MSG}) := \left(1 - \frac{\sum_{k=1}^{|m_{KEYS}|} m_{KEYS}.loss_k + \sum_{k=1}^{|m_{COM}|} m_{COM}.loss_k + \sum_{k=1}^{|m_{REST}|} m_{REST}.loss_k}{|m_{KEYS}| + |m_{COM}| + |m_{REST}|}\right) * \frac{|m_{KEYS}| + |m_{COM}| + |m_{REST}|}{|m_{MSG}.msg_s|}, \text{ where } m_{KEYS}, m_{COM}, m_{REST} \in m_{MSG} \quad (7)$$

$$\mathcal{F}_{PR}(x, y) = \frac{x + x * y}{2} \quad (8)$$

information transparency ($\mathcal{IT}$, the complement of information loss). The metric takes the complement, since these percentages reflect the complement of the translation cost.

We assume constraints have higher priority than information transparency, since constraint violations make the validation of message instances to fail. Thus, information transparency contributes only if the percentages of the satisfied constraints are high. Since both integrity and structural constraints have the same priority, the metric calculates the product of their percentages (Table V (Eq. 4)). Finally, the metric calculates the total message-translation cost using the priority-based function, $\mathcal{F}_{PR}$ (Table V (Eq. 4)), defined in Section IV-B for meeting the previous requirements.

**Metric of satisfied integrity-constraints**. We propose the metric, $\mathcal{INT}$ (Table V (Eq. 5)), which divides the number of the mapped keys of the target message by its keys number.

**Metric of satisfied structural-constraints**. We propose the metric, $\mathcal{STR}$ (Table V (Eq. 6)), which aggregates the percentages of the mapped compulsory and rest elements of the target message. The first (resp. second) percentage is calculated by dividing the number of the mapped compulsory (resp. rest) elements by the compulsory (resp. rest)-elements number. Finally, since both percentages have the same priority, the metric calculates their product.

**Metric of information transparency**. To calculate the percentage of information transparency, we propose the metric, $\mathcal{IT}$ (Table V (Eq. 7)), which aggregates the information-loss percentages at the target and source messages. The first percentage is calculated by taking the complement of the information-loss percentages of the key, compulsory, and rest elements at the target message. The second percentage is calculated by dividing the total number of the mapped elements of the source message by its total elements number.

**Cardinalities of $KEYS$, $COM$, and $REST$ sets**. The cardinality of each set is calculated by adding the max occurrence numbers of all elements of the set.

### *B. Priority-based Aggregation Function*

In $\mathcal{F}_{PR}$ (Table V (Eq. 8)), the prioritized role of the $x$ objective is reflected by the fact that $x$ contributes with its entire value. On the contrary, the non-prioritized role of the $y$ objective is reflected by that fact that $y$ contributes with its product with $x$, which is of lower magnitude order than the $x$ value (given that the objective values belong to [0, 1]).

## V. THE COMPOSITE WORKFLOW OF OUR APPROACH

Fig. 2 depicts the proposed workflow, in which a pair of services is initially pre-processed (determining schema constraints) and represented in our service-interface model. Following, a high-level mapping tool is adopted to produce mappings between service operations and the leaf nodes

of their message schemas, represented in our mapping model. Finally, the workflow uses our metric for estimating the service-translation cost. Since decoupling mechanisms usually adopt service retrieval or clustering approaches to identify similar services, we further describe how our approach can be integrated with such mechanisms.

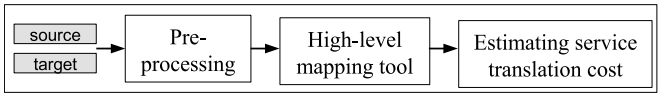



Figure 2. The composite workflow, followed by our approach.

**Service retrieval**. Given a service, whose interface will be used by an adapter, the workflow is iteratively used for comparing the service against a set of candidate services. The top-k similar services are then used as adapted services.

**Service clustering**. The workflow is iteratively used for comparing every pair of services. From the formed clusters, the one whose member services have the total lowest translation cost is selected. Following, the interface of the adapter is identified by the service, whose translation cost to the other services of the selected cluster is min. The remaining services of the cluster are used as adapted services.

## VI. Experimental Evaluation

We implemented in Java service retrieval and clustering mechanisms (Sections VI-B and VI-C), which use our metric and the similarity metric of the approach in [17]. We selected this approach since it provides a high-level mapping tool that examines subtyping relation between services. We evaluate the effectiveness of our approach using the Web services of the `OWLS-TC4`[6] benchmark. The executable file of our research prototype and the evaluation results are available at this location[7]. We firstly set up our experiments.

### A. Experiments Setup

**Benchmark**. `OWLS-TC4` provides the result sets of 42 queries, $Q_1$-$Q_{42}$ (Table VI), and *ideal* high-level mappings.

**Methodology**. Our evaluation includes two parts. In the first part, for each query, we form all the service pairs, we estimate their translation costs and similarity values (using the high-level mapping tool). We also calculate the effectiveness of the high-level mappings. Finally, we randomly generate for each service a number (e.g. 10) of invocation instances and we calculate its average actual translation-costs.

In the second part, since the services number is 312, we use a subset of them, which offers small and cohesive service clusters (Table VII). We adopt the typical (hierarchical bottom-up) clustering method [31], which forms a hierarchy of clusters (called dendrograms). We execute the method twice (once for our metric and once for the similarity metric). In our metric, every service pair is examined in both directions and is included in a cluster if its translation cost is greater than zero in at least one direction.

[5]This query includes only one service with only one interface.

[6]http://projects.semwebcentral.org/projects/owls-tc

[7]ecs.victoria.ac.nz/foswiki/pub/Main/DionysisAthanasopoulos/sources.zip

Table VI
Results on service retrieval for the OWLS-TC4 queries.

| | **#interface pairs** | **Translation Cost vs. Similarity** | | | **Estimated vs. Actual** (close) |
|---|---|---|---|---|---|
| | | lower | close | or higher cost | |
| $Q_1$ | 110 | 50 | 44 | 16 | 110 |
| $Q_2$ | 2 | 1 | 1 | 0 | 2 |
| $Q_3$ | 20 | 0 | 8 | 12 | 20 |
| $Q_4$ | 110 | 36 | 24 | 50 | 110 |
| $Q_5$ | 182 | 86 | 53 | 43 | 182 |
| $Q_6$ | 56 | 26 | 9 | 21 | 52 |
| $Q_7$ | 420 | 118 | 82 | 220 | 420 |
| $Q_8$ | 20 | 7 | 9 | 4 | 20 |
| $Q_9$ | 6 | 0 | 6 | 0 | 6 |
| $Q_{10}$ | 12 | 0 | 4 | 8 | 12 |
| $Q_{11}$ | 6 | 0 | 2 | 4 | 6 |
| $Q_{12}$ | 56 | 2 | 25 | 29 | 56 |
| $Q_{13}$ | 182 | 50 | 46 | 86 | 182 |
| $Q_{14}$ | 20 | 1 | 5 | 14 | 20 |
| $Q_{15}$ | 20 | 4 | 4 | 12 | 20 |
| $Q_{16}$ | 12 | 6 | 2 | 4 | 12 |
| $Q_{17}$ | 420 | 144 | 105 | 171 | 420 |
| $Q_{18}$ | 56 | 8 | 15 | 33 | 56 |
| $Q_{19}$ | 12 | 3 | 3 | 6 | 12 |
| $Q_{20}$ | 306 | 136 | 51 | 119 | 306 |
| $Q_{21}$ | 90 | 24 | 46 | 20 | 90 |
| $Q_{22}$ | 6 | 2 | 1 | 3 | 6 |
| $Q_{23}$ | 56 | 13 | 18 | 25 | 54 |
| $Q_{24}$ | 132 | 41 | 52 | 39 | 132 |
| $Q_{25}$ | 20 | 0 | 8 | 12 | 20 |
| $Q_{26}$ | 342 | 57 | 71 | 178 | 342 |
| $Q_{27}$ | 182 | 32 | 37 | 113 | 182 |
| $Q_{28}$ | 42 | 12 | 21 | 9 | 42 |
| $Q_{29}$ | 272 | 78 | 122 | 72 | 272 |
| $Q_{30}$ | 6 | 1 | 2 | 3 | 6 |
| $Q_{31}$[5] | 0 | 0 | 0 | 0 | 0 |
| $Q_{32}$[5] | 0 | 0 | 0 | 0 | 0 |
| $Q_{33}$ | 6 | 2 | 2 | 2 | 6 |
| $Q_{34}$[5] | 0 | 0 | 0 | 0 | 0 |
| $Q_{35}$ | 12 | 5 | 3 | 4 | 12 |
| $Q_{36}$ | 110 | 40 | 42 | 28 | 90 |
| $Q_{37}$ | 12 | 5 | 4 | 3 | 12 |
| $Q_{38}$ | 20 | 6 | 2 | 4 | 20 |
| $Q_{39}$ | 42 | 17 | 9 | 16 | 36 |
| $Q_{40}$[5] | 0 | 0 | 0 | 0 | 0 |
| $Q_{41}$[5] | 0 | 0 | 0 | 0 | 0 |
| $Q_{42}$[5] | 0 | 0 | 0 | 0 | 0 |

**Mapping effectiveness**. For each pair of mapped operations, we compare their schema elements against the ideally mapped schema-elements. To calculate the overall effectiveness, we adopt the *F-measure* metric [32] (high F-measure indicates high effectiveness).

### B. Evaluation Results on Service Retrieval

The grouped-bar charts of Fig. 3 provide details on the results for two representative queries. The x-axis of the charts correspond to service pairs and the y-axis jointly depicts translation costs, similarity and F-meaure values. We observe that translation cost is not necessarily in accordance to similarity, since in the majority of service pairs, their difference is not low ($\leq 0.2$). When the translation cost is much

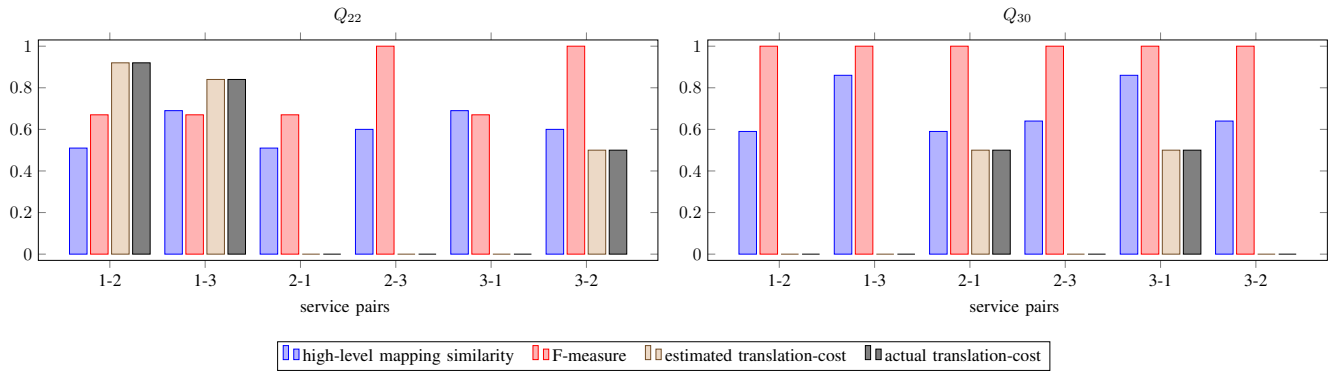



Figure 3. Detailed evaluation results on service retrieval for two OWLS-TC4 queries.

lower than high similarity, it means that services are not only similar, but also there is no integrity or structural constraint-violation. When the translation cost is much higher than high similarity, it means that even if the services are similar, there are constraint violations. Inspecting the results in these cases, we observed that there are not necessarily a lot of violations, but these violations were enough to increase the translation cost (conjunctive nature of our metric).

Another observation is about the directed nature of the translation process, which is also verified from Fig. 3 (different translation costs for the same service pairs in the two directions, e.g. pairs, 1-2 and 2-1, in $Q_{22}$). We also observe that F-measure is high when similarity is also high. Thus, the used high-level mappings are good enough, enabling us to deal with the effectiveness of our approach exclusively. Finally, we observe that the actual costs, depicted in the previous grouped-bar charts (Fig. 3), are very close (usually identical) to the estimated costs (difference lower than 0.1).

The results for all of the queries are included in Table VI, which provides the numbers of service pairs with estimated costs lower, close to, or higher than their similarity, and very close to actual costs. We also calculated the corresponding percentages, which are , 28%, and 42%, and 90%, respectively. The 72% of service pairs has estimated cost much lower or higher of similarity and the 90% very close to the actual cost, verifying our previous main conclusions.

### C. Evaluation Results on Service Clustering

Comparing the dendrograms (Fig. 4), we start from their lowest levels. We observe that the low-level clusters (depicted by rectangles), produced using the service-translation metric (Fig. 4 (b)), are closer to the ideal clusters (Table VII), than those using the similarity metric (Fig. 4 (a)). Inspecting the results, we observed that it is due to (i) the directed nature of service translation and (ii) schema constraints.

Concerning the first reason, the bidirectional nature of service similarity is misleading, since services (e.g. their data-types) were not compatible at both directions. When the aggregation of the two directions is performed, the overall service similarity is reduced (e.g. $i_3$ vs. $i_1$ or $i_2$). Regarding the second reason, even a few schema-constraint violations exist, the estimated cost is significantly reduced. On the contrary, the similarity metric ignores constraint violations, leading to a high overall similarity (e.g. $i_{12}$ vs. $i_1$ or $i_2$). Thus, ensured schema-constraints improve service similarity.

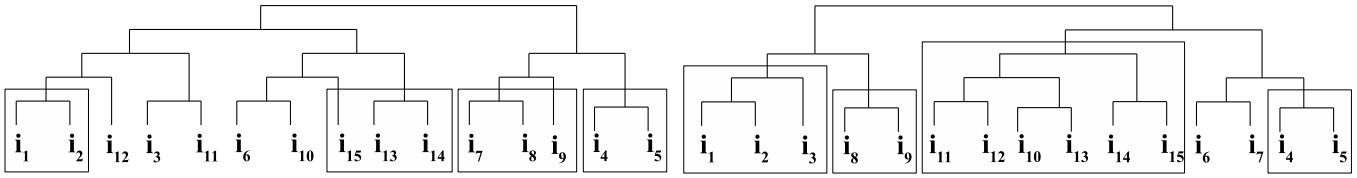
Table VII
THE OWLS-TC4 QUERIES USED BY THE CLUSTERING METHOD.

| | Interfaces | IDs |
|---|---|---|
| $Q_9$ | Ebookorder3Soap | $i_1$ |
| | Ebookorder2Soap | $i_2$ |
| | Ebookorder1Soap | $i_3$ |
| $Q_{11}$ | Geographical-regionWeatherprocessSoap | $i_4$ |
| | Geopolitical-entityWeatherprocessSoap | $i_5$ |
| | Geographical-regionWeatherprocessSoap | $i_6$ |
| $Q_{22}$ | ShoppingmallCalendar-datepricecameraSoap | $i_7$ |
| | ShoppingmallCalendar-datepricecameraSoap | $i_8$ |
| | ShoppingmallCamerapriceSoap | $i_9$ |
| $Q_{30}$ | AltitudeLocationSoap | $i_{10}$ |
| | ElevationLocationSoap | $i_{11}$ |
| | AltitudeLocationSoap | $i_{12}$ |
| $Q_{33}$ | CalculateSunriseSunsetTwilightSoap | $i_{13}$ |
| | CalculateSunriseSunsetSoap | $i_{14}$ |
| | CalculateSunriseSunsetSoap | $i_{15}$ |

## VII. Conclusions and Future Work

We proposed an approach that estimates service-translation cost, reflecting an upper-bound of the satisfied schema constraints. The evaluation results showed that service similarity is not necessarily in accordance to translation cost, the bidirectional nature of similarity can be misleading, ensured schema constraints improve similarity, the estimated and actual translation costs are very close, and service retrieval and clustering are effective using our approach.

Our future focuses on extending the approach to estimate translation cost based on a service-translation mechanism. To this direction, we further need to specify the notion of low-level mappings, their relation to high-level mappings, and a translation mechanism. A final direction is to estimate translation cost at service-composition level.




(a) When the service-similarity metric is used.

(b) When the the service-translation metric is used.

Figure 4. The evaluation results on the service-clustering method.

## REFERENCES


[1] W. Kongdenfha, H. R. M. Nezhad, B. Benatallah, and R. Saint-Paul, "Web service adaptation: Mismatch patterns and semi-automated approach to mismatch identification and adapter development," in *Web Services Foundations*, 2014, pp. 245–272.

[2] P. G. Kolaitis, J. Panttaja, and W. C. Tan, "The complexity of data exchange," in *Symposium on Principles of Database Systems*, 2006, pp. 30–39.

[3] L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernández, and R. Fagin, "Translating web data," in *International Conference on Very Large Data Bases*, 2002, pp. 598–609.

[4] D. Athanasopoulos and A. V. Zarras, "Multi-objective service similarity metrics for more effective service engineering methods," in *IEEE International Conference on Service-Oriented Computing and Applications*, 2015, pp. 208–212.

[5] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.

[6] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity search for web services," in *International Conference on Very Large Data Bases*, 2004.

[7] J. Ma, Q. Z. Sheng, K. Liao, Y. Zhang, and A. H. H. Ngu, "Ws-finder: A framework for similarity search of web services," in *International Conference on Service-Oriented Computing*, 2012, pp. 313–327.

[8] Y. Elshater, K. Elgazzar, and P. Martin, "godiscovery: Web service discovery made efficient," in *International Conference on Web Services*, 2015, pp. 711–716.

[9] C. Li, R. Zhang, J. Huai, X. Guo, and H. Sun, "A probabilistic approach for web service discovery," in *International Conference on Services Computing*, 2013, pp. 49–56.

[10] P. Plebani and B. Pernici, "Urbe: Web service retrieval based on similarity evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 11, 2009.

[11] J. Wu and Z. Wu, "Similarity-based web service matchmaking," in *IEEE International Conference on Services Computing*, 2005, pp. 287–294.

[12] G. Spanoudakis and A. Zisman, "Discovering services during service-based system design using uml," *IEEE Transactions on Software Engineering*, vol. 36, no. 3, pp. 371–389, 2010.

[13] E. Stroulia and Y. Wang, "Structural and semantic matching for assessing web-service similarity," *International Journal of Cooperative Information Systems*, pp. 407–438, 2005.

[14] Y. Hao and Y. Zhang, "Web services discovery based on schema matching," in *Australasian Computer Science Conference*, 2007, pp. 107–113.

[15] V. W. Chu, R. K. Wong, W. Chen, I. Paik, and C. Chi, "Service discovery based on objective and subjective measures," in *International Conference on Services Computing*, 2013, pp. 360–367.

[16] V. Andrikopoulos and P. Plebani, "Retrieving compatible web services," in *International Conference on Web Services*, 2011, pp. 179–186.

[17] D. Athanasopoulos, A. V. Zarras, P. Vassiliadis, and V. Issarny, "Mining service abstractions," in *International Conference on Software Engineering*, 2011, pp. 944–947.

[18] B. Liskov and J. Wing, "A Behavioral Notion of Subtyping," *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 6, pp. 1811–1841, 1994.

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[20] P. Kaminski, H. Muller, and M. Litoiu, "A design for adaptive Web service evolution," in *Software Engineering for Adaptive and Self-Managing Systems*, 2006.

[21] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, "End-to-End Versioning Support for Web Services," in *IEEE International Conference on Services Computing*, 2008.

[22] S. Ponnekanti and A. Fox, "Interoperability Among Independently Evolving Web Services," in *International Middleware Conference*, 2004.

[23] S. R. Ponnekanti, "Application-Service Interoperation Without Standardized Service Interfaces," in *International Conference on Pervasive Computing and Communications*, 2003.

[24] H. R. M. Nezhad, B. Benatallah, A. Martens, F. Curbera, and F. Casati, "Semi Automated Adaptation of Service Interactions," in *International World Wide Web Conference*, 2007, pp. 993–1002.

[25] L. Cavallaro and E. D. Nitto, "An approach to adapt service requests to actual service interfaces," in *Software Engineering for Adaptive and Self-Managing Systems*, 2008, pp. 129–136.

[26] D. Athanasopoulos, A. Zarras, and V. Issarny, "Service substitution revisited," in *Automated Software Engineering*, 2009, pp. 555–559.

[27] L. Melloul and A. Fox, "Reusable Functional Composition Patterns for Web Services," in *IEEE International Conference on Web Services*, 2004.

[28] J. Yang and M. Papazoglou, "Service Components for Managing the Lifecycle of Service Compositions," *Information Systems*, vol. 29, no. 2, pp. 97–125, 2004.

[29] Y. Taher, D. Benslimane, M.-C. Fauvet, and Z. Maamar, "Towards an Approach for Web Services Substitution," in *International Database Engineering and Applications Symposium*, 2006.

[30] X. Liu and H. Liu, "Automatic abstract service generation from web service communities," in *International Conference on Web Services*, 2012, pp. 154–161.

[31] O. Maqbool and H. A. Babri, "Hierarchical clustering for software architecture recovery," *IEEE Transactions on Software Engineering*, vol. 33, no. 11, pp. 759–780, 2007.

[32] R. A. Baeza-Yates and B. A. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press/Addison-Wesley, 1999.